

establishing the microprocessor 210 as a slave and the test controller 120 as a master. Embodiments described herein provide further details of test mode entry methodology.

5 In step 620, the test controller 120 fills the instruction queue 240 with instructions to be executed in the microprocessor 210. These instructions may originate from the test interface 110. By originating from the test interface 110 it may be meant that the instruction that the test controller 120 transfers to the instruction queue 240 is based on a command or instruction received over the test
10 interface 110. However, the test controller 120 may provide some translation or modification of the received instruction. Thus, it is not required that there be a one-to-one correspondence between the instruction received over the test interface 120 and the one sent to the instruction queue 240.

15 In step 630, the microprocessor 210 is forced to execute instructions from the instruction queue 240. In this fashion, the program flash 260 is bypassed when the circuit 150 is in test mode. If desired, a string of instructions may be single stepped in this fashion.

20 In optional step 640, the test controller 120 transfers an instruction to the instruction queue 240 that, when received by the microprocessor 210, cause the microprocessor 210 to execute instructions from the supervisory ROM 250 instead. This may be referred to as a supervisory or privileged state, in which greater access to various registers is provided for more complete testing.

In step 650, after a set of instructions from the supervisory ROM 250 completes, the next instruction in the instruction queue 240 is executed. The process 600 may continue by executing more instructions from both the instruction queue 240 and the SROM 250, until the test mode is left, in step 660.

Figure 5 shows an embodiment of a process 700 in which a single instruction sent to the instruction queue 240 may cause, in effect, a sub-routine call to execute a set of instructions in the supervisory ROM 250. After establishing the test mode in step 710, the test controller 120 transfers an instruction the instruction queue 240, in step 720.

In step 730, while still in test mode, a supervisory state is entered, in which the microprocessor 210 executes instructions from the supervisory ROM 250 instead of the instruction queue 240. In this state, the microprocessor may take control of the multiplexer 270 and cause instructions from the SROM 250 to be fed into it, although this is not required. While in supervisory state it may be possible to test the circuit 150 at normal speed (e.g., at 100 percent clock speed).

In optional step 740, various registers are tested by writing to a register during the supervisory state. The given register is one that is not writeable during the test mode when not in supervisory state. Thus, it may be possible to test elements during test mode when in supervisory state that may not be fully testable during test mode when not in supervisory state.

In step 750, the supervisory state is left, and instructions are once again executed from the instruction queue 240. In one embodiment, the microprocessor 210, which had taken control of the multiplexer 270 during the supervisory state, allows the test controller 120 to once again control the multiplexer 270.

In optional step 760, the process continues to switch between executing instructions from the instruction queue 240 and the supervisory ROM 250, all while in test mode. In this fashion, the microprocessor 210 switches between executing instructions originating from the test interface 110 and pre-determined instructions from the SROM 250. The process 700 may then end and test mode exited.

It is appreciated that various steps in the embodiments illustrated in process 600 and process 700 of Figures 4 and 5 may be optional and that steps from one process may be used in the other.

TEST MODE ENTRY METHODOLOGY

The following describes embodiments which provide for a method to enter test mode of circuit 150. When voltage is applied to the circuit 150 as part of its normal power on, a special negotiation takes place over the test interface 110 during a window of time after power on. Furthermore, a password may be required. Beneficially, the negotiation does not interfere with chip performance.

Figure 6 represents traces indicating the sequence of events during negotiation of entry into test mode. It will be understood that while some of the